CLAIMS

1 1. A locking system implemented on a distributed file system where clients directly access data

2 on storage devices via a storage area network and a file server provides metadata for said data and

3 manages revocation and granting of locks of said lock system, said lock system comprising:

4 a consumer lock, said consumer lock allowing a reader granted said consumer lock

5 to read a block of data;

6 a producer lock, said producer lock allowing a writer granted said producer lock to

7 change said block of data, and

8 wherein completion of said writer changing said block of data, said writer releases

9 said producer lock, and upon release of said producer lock, said reader granted said consumer lock

10 is notified said block of data has changed.

1 2. A locking system implemented on a distributed file system where clients directly access data

2 on storage devices via a storage area network and a file server provides metadata for said data and

3 manages revocation and granting of locks of said lock system as per claim 1, wherein said block of

4 data is changed by writing updated data to a physical storage location different than where said block

5 of data is stored.

1   3.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 2, wherein said

4   notification informs said reader of said updated data location.

1   4.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 3, wherein said

4   notification causes a cached copy of said block of data held in a cache of said reader to be

5   invalidated.

1   5.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 2, wherein reads

4   performed on said block of data by said reader after receiving said notification are performed by

5   reading said updated data from said updated data location.

1   6.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 2, wherein said reader

4   continues to read said block of data from the physical storage location said block of data is stored

5   while said writer is writing said updated data to said different physical storage location.

1   7.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 1, wherein said writer

4   writes data to storage devices physically separated from a storage device located on said file system

5   server.

1   8.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 7, wherein said writer

4   writes data to said physically separate storage devices via a storage area network.

1   9.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 7, wherein said storage

4   device located on said file system server stores metadata.

1   10.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 7, wherein said physically

4   separate storage devices cache data for read operations.

1   11.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 1, wherein said reader is

4   a web server.

1   12.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 1, wherein said writer is

4   a database management system.

1   13.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 1, wherein said reader

4   comprises a plurality of readers, each of said readers being granted separate consumer locks.

1   14.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 1, wherein said lock

4   system is implemented on a system where said reader and said writer access data directly from

5   storage devices via a storage area network and said readers and said writers access metadata from

6   said file server via a data network separate from said storage area network.

1   15.   A locking system implemented on a distributed file system where clients directly access data

2   on storage devices via a storage area network and a file server provides metadata for said data and

3   manages revocation and granting of locks of said lock system as per claim 1, wherein said lock

4   system is implemented in a distributed file system which utilizes multiple locking systems for data

5   where the locking system used for a particular block of data is dependent on what application utilizes

6   said particular block of data and the locking system utilized for the particular block of data is

7   indicated by the metadata corresponding to said particular block of data.

1   16.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block,

4   said method comprising:

5   receiving a request from said writer to grant said producer lock to said writer;

6   granting said producer lock to said writer;

7   receiving a producer lock release message, said producer lock release message being

8   received after said writer completes updating of said data block;

9   sending an update message to said readers holding said consumer lock, said update

10   message notifying said readers said data block has been updated.

11   17.   A method of updating a data block in a distributed file system including a consumer lock,

12   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

13   producer lock, said producer lock granted to a writer to allow said writer to update said data block

14   as per claim 16, wherein said data block is updated by writing updated data to a different physical

15   storage location than where said data block is stored.

1   18.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 17, wherein said update message informs said readers of said updated data location.

1   19.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 18, wherein said update message causes a cached copy of said data block held in a cache

5   of said readers to be invalidated.

1   20.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 17, wherein reads performed on said data block by said readers after receiving said

5   update message are performed by reading said updated data from said updated data location.

1   21.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 17, wherein said reader continues to read said data block from the physical storage

5   location said data block is stored while said writer is writing said updated data to said different

6   physical storage location.

1   22.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 16, wherein said writer writes data to storage devices physically separated from a

5   storage device located on said file system server.

1   23.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 22, wherein said writer writes data to said physically separate storage devices via a

5   storage area network.

1   24.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 22, wherein said storage device located on said file system server stores metadata.

1   25.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 22, wherein said physically separate storage devices cache data for read operations.

26.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 16, wherein said reader is a web server.

1   27.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 16, wherein said writer is a database management system.

1   28.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 16, wherein said lock system is implemented on a system where said readers and said

5   writer access data directly from storage devices via a storage area network and said readers and said

6   writers access metadata from said file server via a data network separate from said storage area

7   network.

29.   A method of updating a data block in a distributed file system including a consumer lock,

said consumer lock granted to multiple readers to allow said readers to read said data block, and a

producer lock, said producer lock granted to a writer to allow said writer to update said data block

as per claim 16, wherein said method is implemented in a distributed file system which utilizes

multiple locking systems for data where the locking system used for a particular block of data is

dependent on what application utilizes said particular block of data and the locking system utilized

7   for the particular block of data is indicated by the metadata corresponding to said particular block

8   of data.

1   30.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block,

4   said method comprising:

5   sending a request for said producer lock;

6   receiving said producer lock;

7   updating said data block;

8   releasing said producer lock after said updating is completed.

1   31.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 30, said method further comprising:

5   sending an update message to said readers granted said consumer lock after said

6   releasing step, said update message notifying said readers said data block has been updated.

1   32.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 30, wherein said updating step comprises writing updated data to a different physical

5   storage location than where said data block is stored.

1   33.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 31, wherein said updating step comprises writing updated data to a different physical

5   storage location than where said data block is stored.

1   34.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 33, wherein said notification informs said readers of said updated data location.

1   35.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 34, wherein said notification causes a cached copy of said data block held in a cache of

5   said readers to be invalidated.

1   36.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 33, wherein reads performed on said data block by said readers after receiving said

5   notification are performed by reading said updated data from said updated data location.

1   37.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 33, wherein said readers continue to read said data block from the physical storage

5   location said block of data is stored while said writer is writing said updated data to said different

6   physical storage location.

1   38.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 30, wherein said writer writes data to storage devices physically separated from a

5   storage device located on said file system server.

1   39.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block,

4   as per claim 38, wherein said writer writes data to said physically separate storage devices via a

5   storage area network.

1   40.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 30, wherein said storage device located on said file system server stores metadata.

1   41.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 38, wherein said physically separate storage devices cache data for read operations.

1   42.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 30, wherein said reader is a web server.

1   43.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 30, wherein said writer is a database management system.

1   44.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 30, wherein said method is implemented on a system where said readers and said writer

5   access data directly from storage devices via a storage area network and said readers and said writers

6   access metadata from said file server via a data network separate from said storage area network.

1   45.   A method of updating a data block in a distributed file system including a consumer lock,

2   said consumer lock granted to multiple readers to allow said readers to read said data block, and a

3   producer lock, said producer lock granted to a writer to allow said writer to update said data block

4   as per claim 30, wherein said method is implemented in a distributed file system which utilizes

5   multiple locking systems for data where the locking system used for a particular block of data is

6   dependent on what application utilizes said particular block of data and the locking system utilized

7   for the particular block of data is indicated by the metadata corresponding to said particular block

8   of data.

1   46.   A system for providing multiple quality of service options through locking such that a file

2   system handling data consistency and cache coherency is useful to a variety of application types, said

3   system comprising:

4   a plurality of locking protocols, each of said locking protocols designed to support

5   a different application's data consistency and cache coherency needs;

6   wherein one of said plurality of locking protocols is assigned to a data segment based

7   upon which application utilizes said data segment.

47.   A system for providing multiple quality of service options through locking such that a file

system handling data consistency and cache coherency is useful to a variety of application types, as

per claim 46, wherein one of said locking protocols implements a sequential consistency model

including file system caching.

48.   A system for providing multiple quality of service options through locking such that a file

2   system handling data consistency and cache coherency is useful to a variety of application types, as

3   per claim 46, wherein one of said locking protocols implements a sequential consistency model

4   without file system caching.

1 49. A system for providing multiple quality of service options through locking such that a file

2 system handling data consistency and cache coherency is useful to a variety of application types, as

3 per claim 46, wherein one of said locking protocols implements a weak consistency model including

4 file system caching.

1 50. A system for providing multiple quality of service options through locking such that a file

2 system handling data consistency and cache coherency is useful to a variety of application types, as

3 per claim 49, wherein said locking protocol which implements a weak consistency model including

4 file system caching comprises:

5   a consumer lock, said consumer lock allowing a reader granted said consumer lock

6 to read a block of data;

7   a producer lock, said producer lock allowing a writer granted said producer lock to

8 change said block of data, and

9   wherein completion of said writer changing said block of data, said writer releases

10 said producer lock, and upon release of said producer lock, said reader granted said consumer lock

11 is notified said block of data has changed.

1   51.   A system for providing multiple quality of service options through locking such that a file

2   system handling data consistency and cache coherency is useful to a variety of application types, as

3   per claim 46, wherein said multiple locking protocols comprise:

4   a locking protocol which implements a sequential consistency model including file

5   system caching;

6   a locking protocol which implements a sequential consistency model without file

7   system caching, and

8   a locking protocol which implements a weak consistency model including file system

9   caching.

1   52.   A system for providing multiple quality of service options through locking such that a file

2   system handling data consistency and cache coherency is useful to a variety of application types, as

3   per claim 51, wherein said locking protocol which implements a weak consistency model including

4   file system caching comprises:

5   a consumer lock, said consumer lock allowing a reader granted said consumer lock

6   to read a block of data;

7   a producer lock, said producer lock allowing a writer granted said producer lock to

8   change said block of data, and

9   wherein completion of said writer changing said block of data, said writer releases

10   said producer lock, and upon release of said producer lock, said reader granted said consumer lock

11   is notified said block of data has changed.

53.   A system for providing multiple quality of service options through locking such that a file

system handling data consistency and cache coherency is useful to a variety of application types, as

3   per claim 46, wherein said locking protocol assigned to a data segment is assigned based upon

4   default locking protocols being set for portions of the file system name space.

1   54.   A system for providing multiple quality of service options through locking such that a file

2   system handling data consistency and cache coherency is useful to a variety of application types, as

3   per claim 46, wherein said locking protocol assigned to a data segment is assigned based upon any

4   of owner, group or file extension.

1   55.   A system for providing multiple quality of service options through locking such that a file

2   system handling data consistency and cache coherency is useful to a variety of application types, as

3   per claim 46, wherein said locking protocol assigned to a data segment is changed by a file system

4   call.

1   56.   A system for providing multiple quality of service options through locking such that a file

2   system handling data consistency and cache coherency is useful to a variety of application types, as

3   per claim 46, wherein a client accessing said data segment determines said locking protocol assigned

4   to said data segment via file metadata.

1   57.   A system for providing multiple quality of service options through locking such that a file

2   system handling data consistency and cache coherency is useful to a variety of application types, as

3   per claim 46, wherein a client accessing said data segment determines said locking protocol assigned

4   to said data segment via a file extension.

1   58.   A distributed computing system including a file system handling cache coherency and data

2   consistency providing multiple quality of service through locking protocols, said system comprising:

3   a server, said server connected to at least one client of said distributed computing

4   system via a first data network, said server serving file metadata to said client upon said client

5   accessing a file stored in said distributed computing system, said server managing data consistency

6   and cache coherency through multiple locking protocols designed to support different application's

7   data consistency and cache coherency needs;

8   a storage device connected to said client via a second data network, said storage

9   device storing file data;

10   wherein one of said multiple locking protocols is assigned to a file based upon which

11   application utilizes said file and said locking protocol assigned to said file is indicated via said file

12   metadata.

1   59.   A distributed computing system including a file system handling cache coherency and data

2   consistency providing multiple quality of service through locking protocols, as per claim 58, wherein

3   said multiple locking protocols comprise:

4   a locking protocol which implements a sequential consistency model including file

5   system caching;

6   a locking protocol which implements a sequential consistency model without file

7   system caching, and

8   a locking protocol which implements a weak consistency model including file system

9   caching.

1   60.   A distributed computing system including a file system handling cache coherency and data

2   consistency providing multiple quality of service through locking protocols, as per claim 58, wherein

3   said at least one client comprises a plurality of clients and said locking protocol which implements

4   a weak consistency model including file system caching comprises:

5   a consumer lock, said consumer lock allowing one of said plurality of clients, acting

6   as a reader, granted said consumer lock to read a block of data;

7   a producer lock, said producer lock allowing one of said clients, acting as a writer,

8   granted said producer lock to change said block of data, and

9   wherein completion of said writer changing said block of data, said writer releases

10   said producer lock, and upon release of said producer lock, said reader granted said consumer lock

11   is notified said block of data has changed.

1   61.   A distributed computing system including a file system handling cache coherency and data

2   consistency providing multiple quality of service through locking protocols, as per claim 60, wherein

3   said block of data is changed by writing updated data to a physical storage location different than

4   where said block of data is stored.

1   62.   A distributed computing system including a file system handling cache coherency and data

2   consistency providing multiple quality of service through locking protocols, as per claim 61, wherein

3   said notification informs said reader of said updated data location.

1 63. A distributed computing system including a file system handling cache coherency and data

2 consistency providing multiple quality of service through locking protocols, as per claim 62, wherein

3 said notification causes a cached copy of said block of data held in a cache of said reader to be

4 invalidated.

1 64. A distributed computing system including a file system handling cache coherency and data

2 consistency providing multiple quality of service through locking protocols, as per claim 61, wherein

3 reads performed on said block of data by said reader after receiving said notification are performed

4 by reading said updated data from said updated data location.

1 65. A distributed computing system including a file system handling cache coherency and data

2 consistency providing multiple quality of service through locking protocols, as per claim 61, wherein

3 said reader continues to read said block of data from the physical storage location said block of data

4 is stored while said writer is writing said updated data to said different physical storage location.

1 66. A distributed computing system including a file system handling cache coherency and data

2 consistency providing multiple quality of service through locking protocols, as per claim 60, wherein

3 said reader is a web server.

1   67.   A distributed computing system including a file system handling cache coherency and data

2   consistency providing multiple quality of service through locking protocols, as per claim 60, wherein

3   said writer is a database management system.

1   68.   A distributed computing system including a file system handling cache coherency and data

2   consistency providing multiple quality of service through locking protocols, as per claim 60, wherein

3   said reader comprises a plurality of readers, each of said readers being granted separate consumer

4   locks.